

CLAIM AMENDMENT

Please amend the claims as follows:

1. (Canceled)
2. (Canceled)
3. (Canceled)
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Canceled)
8. (currently amended) A method for using a first computer system to remotely monitor and interact with the operation of a second computer system through a graphical user interface of said second computer system, comprising the steps of:

receiving a pixel image of said second computer system graphical user interface at
5 said first computer system;

searching said pixel image of said second computer system graphical user interface for
a first graphical element which may be found at one of a plurality of potential
locations contained within and comprising less than said pixel image;

generating a user peripheral input device input action within said second computer
10 system graphical user interface as interpreted by a second computer peripheral
input device controller channel by passing a signal through an i/o
communications channel from said first computer system to said second
computer system graphical user interface responsive to said receiving and
searching steps;

15 monitoring said pixel image of said second computer system graphical user interface
from said first computer system for an expected second graphical element
contained within and comprising less than said pixel image within a
predetermined time interval; and

signaling a failure at said first computer system if said predetermined time interval
20 elapses without detecting said expected second graphical element.

9. (Previously presented) The method of claim 8 further comprising the steps of:

transferring said user input action to a script stored on said first computer system;

re-executing said steps of receiving, generating, monitoring and signaling subsequent to said storing step under control of said stored script.

10. (Previously presented) The method of claim 8 further comprising the steps of:

providing graphical user interface language extensions commands to a scripting language; and

passing said generated user input action through said graphical user interface language extensions from a scripting language processor to a language extensions processor.

11. (currently amended) The method of claim 8 further comprising the steps of:

generating a user input action within said second computer system responsive to said second graphical element;

monitoring said second computer system graphical user interface for an expected third graphical element which may be found at one of a plurality of potential locations contained within and comprising less than said pixel image within a predetermined time interval; and

signaling a failure at said first computer system if said predetermined time interval elapses without detecting said expected third graphical element.

12. (Previously presented) The method of claim 8 further comprising the steps of:
- depicting said second computer system graphical user interface upon a local display
 - of said first computer system including said first graphical element; and
 - receiving a local user input action at said first computer system within said local display;
 - wherein said generated user input action emulates said local user input action.
13. (Previously presented) The method of claim 8 further comprising the steps of:
- providing graphical user interface language extensions commands to a scripting language; and
 - depicting said computer system graphical user interface upon a local display of said first computer system including said first graphical element;
 - receiving a local user input action within said local display;
 - transferring said user input action to a script stored on said first computer system;
 - passing said generated user input action through said graphical user interface language extensions from a scripting language processor to a language extensions processor for reproduction at said second computer system graphical user interface, wherein said generated user input action emulates said local user input action; and
 - re-executing said steps of receiving, searching, generating, monitoring and signaling

subsequent to said storing step under control of said stored script.

14. (currently amended) A method for enabling a local system to remotely operate a remote computer system through a graphical user interface on said remote computer system by using local scripts that selectively respond to changes in graphical displays upon said graphical user interface of said remote computer system, comprising the steps of:

5 displaying a depiction of said remote system graphical user interface display on said
 local system;
 capturing user input effected in said depiction of said remote system graphical user
 interface display;
 implementing through a local system command language set user input emulations
10 representative of said captured user input reproduced at said remote computer
 system graphical user interface through a peripheral input device i/o channel;
 image processing said remote computer system graphical displays to detect a first
 entity which may be found at one of a plurality of possible locations contained
 within and comprising less than said graphical display upon said graphical user
15 interface of said remote computer system;
 controlling a flow of execution of said local system through a scripting language
 having scripting commands in combination with said command language set
 responsive to a detection of said first entity during said image processing step;

and

20 communicating between said local system and said remote computer system graphical user interface through a communication interface responsive to said flow controlling step.

15. (Previously presented) The method for enabling a local system to remotely operate a remote computer system through a graphical user interface on said remote computer system of claim 14 further comprising the steps of:

storing said scripting commands into a storing means;
inserting a command from said command language set into said storing means; and
executing said inserted stored command.

16. (currently amended) A method for using a first computer system to remotely monitor and interact with the operation of a second computer system through a graphical user interface of said second computer system, comprising the steps of:

5 receiving a representation of said second computer system graphical user interface at said first computer system;
searching said representation of said second computer system graphical user interface for a first graphical entity which may be found at one of a plurality of potential locations contained within and comprising less than said representation of said

second computer system graphical user interface;

10 generating a user peripheral input device input action within said second computer
 system graphical user interface as interpreted by a second computer peripheral
 input device controller channel by passing a signal through an i/o
 communications channel from said first computer system to said second
 computer system graphical user interface responsive to said receiving and
15 searching steps;
 monitoring said second computer system graphical user interface from said first
 computer system for an expected second graphical entity within a
 predetermined time interval; and
 signaling a failure at said first computer system if said predetermined time interval
20 elapses without detecting said expected second graphical entity.

17. (Previously presented) The method of claim 16 further comprising the steps of:

 transferring said user input action to a script stored on said first computer system;
 re-executing said steps of receiving and monitoring subsequent to said storing step
 under control of said stored script.

18. (Previously presented) The method of claim 16 further comprising the steps of:

 providing graphical user interface language extensions commands to a scripting

language; and

passing said generated user input action through said graphical user interface language extensions from a scripting language processor to a language extensions processor.

19. (Previously presented) The method of claim 16 further comprising the steps of:

generating a user input action within said second computer system responsive to said second graphical entity;

monitoring said second computer system graphical user interface for an expected third graphical entity within a predetermined time interval; and

signaling a failure at said first computer system if said predetermined time interval elapses without detecting said expected third graphical entity.

20. (Previously presented) The method of claim 16 further comprising the steps of:

depicting said second computer system graphical user interface upon a local display of said first computer system including said first graphical entity; and

receiving a local user input action at said first computer system within said local display;

wherein said generated user input action emulates said local user input action.

21. (Previously presented) The method of claim 16 further comprising the steps of:
- providing graphical user interface language extensions commands to a scripting language; and
 - depicting said computer system graphical user interface upon a local display of said first computer system including said first graphical entity;
 - receiving a local user input action within said local display;
 - transferring said user input action to a script stored on said first computer system;
 - passing said generated user input action through said graphical user interface language extensions from said a scripting language processor to a language extensions processor for reproduction at said second computer system graphical user interface, wherein said generated user input action emulates said local user input action; and
 - re-executing said steps of receiving, generating, monitoring and signaling subsequent to said storing step under control of said stored script.
22. (Previously presented) The method of claim 8, wherein said step of generating a user peripheral input device input action further comprises locating said user peripheral input device input action within said first graphical element.
23. (Previously presented) The method of claim 22, wherein said user peripheral input device

input action further comprises a click event.

24. (Previously presented) The method of claim 14, wherein said step of communicating between said local system and said remote computer system graphical user interface further comprises locating a user peripheral input device input action at a location relative to said first entity.
25. (Previously presented) The method of claim 24, wherein said user peripheral input device input action further comprises a click event.
26. (Previously presented) The method of claim 16, wherein said step of generating a user peripheral input device input action further comprises locating said user peripheral input device input action within said first graphical entity.
27. (Previously presented) The method of claim 26, wherein said user peripheral input device input action further comprises a click event.
28. (New) A method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine by using scripts that selectively respond to changes within graphical displays upon said graphical user interface of said second computing machine, comprising the steps of:

5 displaying a depiction of said second computing machine graphical user interface
display on a graphical user interface of said first computing machine;
capturing user input effected in said depiction of said second computing machine
graphical user interface display;
image processing said second computing machine graphical displays to detect
10 presence of a first entity which may be found at one of a plurality of potential
locations contained within and comprising less than said graphical display
upon said graphical user interface of said second computing machine;
controlling a flow of execution of said first computing machine through a scripting
language having scripting commands in combination with said command
15 language set responsive to a detection of said first entity during said image
processing step;
implementing user input emulations representative of said captured user input
reproduced at said second computing machine graphical user interface
through a peripheral input device i/o channel responsive to said first entity
20 presence; and
communicating between said first computing machine and said second computing
machine graphical user interface through a communication interface
responsive to said flow controlling step.

29. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said step of implementing user input emulations further comprises locating user input emulations at a location determined relative to said first entity.
30. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 29 wherein said step of implementing user input emulations further comprises locating user input emulations directly upon said first entity.
31. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises an icon.
32. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises a graphical control.
33. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of

claim 28 wherein said first entity further comprises a prompt.

34. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises a command button.
35. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises a message box.
36. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said second computing machine further comprises a virtual network computing component.